Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (original) A method for an arithmetic encoding and decoding of binary states,

characterized in that

in a first step a presetable value range for the specification of the interval width R is separated in K representative interval widths $\{Q_1, \ldots, Q_K\}$, a presetable value range for the specification of the probabilities is separated in N representative probability states $\{P_1, \ldots, P_N\}$ and allocation regulations are given, which allocate one Q_K ($1 \leq k \leq K$) to every interval width R and one P_n ($1 \leq n \leq N$) to every probability, and that in a second step the encoding or decoding of the binary states take place by performing the calculation of the new interval width to be derived in the encoding or decoding process, respectively, using a representative interval width Q_K ($1 \leq k \leq K$) and a representative probability state P_n ($1 \leq n \leq N$) by arithmetic operations other than multiplication and division, wherein the representative interval width Q_K is determined by the basic basis interval of the width R and the representative probability state P_n is determined by the probability estimation underlying the symbol to be encoded or to be decoded according to the given allocation regulations.

2. (original) The method according to claim 1,

characterized in that

based on the interval currently to be evaluated having a width R, for determining the associated interval width Q_K, an index q_index is determined by a shift and bit masking operation applied to the computer-internal/binary representation of R.

3. (original) The method according to claim 1,

characterized in that

based on the interval currently to be evaluated with a width R, for the determination of the associated interval width Q_K, an index q_index is determined by a shift operation applied to the computer-internal/-binary representation of R and a downstream access to a table Qtab, wherein the table Qtab contains the indices of interval widths corresponding to values of R prequantized by a shift operation.

4. (original) The method according to claim 1,

characterized in that

the probability estimation underlying the symbol to be encoded or to be decoded is associated with a probability state P_n with the help of an index p_state.

5. (original) The method according to claim 1,

characterized in that

the values of the interval width R_{LPS} corresponding to all K interval widths and to all N different probability states are entered into a table Rtab as product values (Q_K * P_n).

6. (original) The method according to claim 1,

characterized in that

the determination of the interval width R_{LPS} corresponding to the LPS is performed by an access to a table Rtab, wherein the table Rtab contains the values of the interval width R_{LPS} corresponding to all K quantized values of R and to the N different probability states as product values ($Q_K * P_n$).

7. (original) The method according to claim 1,

characterized in that

the determination of the interval width R_{LPS} corresponding to the LPS is performed by an access to the table Rtab, wherein, for an evaluation of the table, the quantization index q_index and the index of the probability state p_state are used.

8. (original) The method according to claim 1,

characterized in that

for the N different representative probability states transition rules are preset, wherein the transition rules indicate which new state is used for the next symbol to be encoded or to be decoded based on the currently encoded or decoded symbol.

9. (original) The method according to claim 8,

characterized in that

a table Next_State_LPS is created which contains the index m of the new probability state P_m for the index n of the currently given probability state P_n at the occurrence of a least probable symbol (LPS).

10. (original) The method according to claim 8,

characterized in that

a table Next_State_MPS is created which contains the index m of the new probability state P_m for the index n of the currently given probability state P_n at the occurrence of a most probable symbol (MPS).

11. (original) The method according to claim 1,

characterized in that

the number K of quantization values and/or the number N of the representative states are selected depending on the preset accuracy of the coding and/or depending on the available storage room.

12. (currently amended) The method according to claim 1,

characterized in that

the table-aided encoding includes the following steps:

~~6.~~f) Determination of the LPS

~~7.~~g) Quantization of R:

q_index = Qtab[R>>q]

~~8.~~h) Determination of R_{LPS} and R:

R_{LPS} = Rtab [q_index, p_state]

R = R - R_{LPS}

~~9.~~i) Calculation of the new partial interval:

if (bit = LPS) **then**

L ← L + R

R ← R_{LPS}

if (p_state = 0) then valMPS ← 1 - valMPS

p_state ← Next_State_LPS [p_state]

else

p_state ← Next_State_MPS [p_state]

~~10.~~j) Renormalization of L and R, writing bits, wherein

q_index describes the index of a quantization value read out of Qtab,

p_state describes the current state,

R_{LPS} describes the interval width corresponding to the LPS and

valMPS describes the bit corresponding to the MPS.

13. (currently amended) The method according to claim 1,

characterized in that

a table-aided decoding includes the following steps:

~~1.~~a) Determination of the LPS

~~2.~~b) Quantization of R:

q_index = Qtab[R>>q]

~~3.~~c) Determination of R_{LPS} and R:

R_{LPS} = Rtab [q_index, p_state]

R = R - R_{LPS}

~~4.~~d) Determination of bit depending on the position of the partial interval:

```
if (V ≥ R) then
    bit ← LPS
    V ← V - R
    R ← R_LPS
    if (p_state = 0) then valMPS ← 1 - valMPS
    p_state ← Next_State_LPS [p_state]
else
    bit ← MPS
    p_state ← Next_State_MPS [p_state]
```

~~5.~~e) Renormalization of R, reading out one bit and updating V, wherein

q_index describes the index of a quantization value read out of Qtab,

p_state describes the current state,

R_{LPS} describes the interval width corresponding to the LPS,

valMPS describes the bit corresponding to the MPS, and

V describes a value from the interior of the current partial interval.

14. (original) The method according to claim 1,

characterized in that

in encoding and/or decoding the calculation of the quantization index q_index is performed in the second substep according to claim 12 and/or 13 according to the calculation regulation:

q_**index** = (R >> q) & Qmask

wherein Qmask illustrates a bit mask suitably selected depending on K.

15. (currently amended) The method according to claim_1,

characterized in that

when a uniform probability distribution is present

in the encoding according to claim 12 the substeps [[~~1~~]] f to [[~~4~~]] i are performed according to the following calculation regulation:

R ← R >> 1

if (bit = 1) **then**

  L ← L + R

or

that the substeps [[~~1~~]] f to [[~~4~~]] i of the encoding according to claim 12 are performed according to the following calculation regulation:

L ← L << 1

if (bit = 1) **then**

L ← L + R

and wherein in the last alternative the renormalization (substep [[~~5~~]] j according to claim 12) is performed with doubled decision threshold values and no doubling of L and R is performed, and

that in the decoding according to claim 13 the substeps [[~~1~~]] a to [[~~4~~]] d are performed according to the following calculation regulation:

R ← R >>1
if (V ≥ R) **then**
  bit ← 1
  V ← V - R
else
  bit ← 0,

or

the substeps [[~~1~~]] a to [[~~5~~]] e of the decoding according to claim 13 are performed according to the following calculation regulation:

~~3.~~m) Reading out one bit and updating V

~~4.~~n) Determination of bit according to the position of the partial interval:

if (V ≥ R) **then**
  bit ← 1
  V ← V - R
else
  bit ← 0.

16. (original) The method according to claim 1,

characterized in that

the initialization of the probability models is performed depending on a quantization parameter SliceQP and preset model parameters m and n, wherein SliceQP describes the

quantization parameter preset at the beginning of a slice and m and n describe the model parameters.

17. (currently amended) The method according to claim 1,

characterized in that

the initialization of the probability models includes the following steps:

~~1.~~k) preState = min(max(1, ((m * SliceQP) >>4)+n), 2*N)
~~2.~~l) **if** (preState <=N) **then**

    p_state = N ~~+1~~ - preState
    valMPS = 0

  else

    p_state = preState - (N+1)
    valMPS = 1,

wherein valMPS describes the bit corresponding to the MPS, SliceQP describes the quantization parameter preset at the beginning of a slice and m and n describe the model parameters.

18. (original) The method according to claim 1,

characterized in that

the probability estimation of the states is performed using a finite state machine (FSM).

19. (original) The method according to claim 1,

characterized in that

the generation of the representative states is performed offline.

20. (original) The method according to claim 1,

characterized in that

the selection of the states depends on the statistics of the data to be coded and/or on the number of states.

21. (original) An arrangement having at least one processor and/or chip, which is/are implemented such that a method for an arithmetic encoding and decoding of binary states is may be performed, wherein

in a first step a presetable value range for the specification of the interval width R is separated in K representative interval widths $\{Q_1, \ldots, Q_K\}$, a presetable value range for the specification of the probabilities is separated in N representative probability states $\{P_1, \ldots, P_N\}$ and allocation regulations are given, which allocate one Q_K $(1 \leq k \leq K)$ to every interval width R and one P_n $(1 \leq n \leq N)$ to every probability, and wherein in a second step the encoding or decoding of the binary states take place by performing the calculation of the new interval width to be derived in the encoding or decoding process, respectively, using a representative interval width Q_K $(1 \leq k \leq K)$ and a representative probability state P_n $(1 \leq n \leq N)$ by arithmetic operations other than multiplication and division, wherein the representative interval width Q_K is determined by the basic basis interval of the width R and the representative probability state P_n is determined by the probability estimation underlying the symbol to be encoded or to be decoded according to the given allocation regulations.

22. (original) A computer program which enables a computer after it has been loaded into the storage of the computer to perform a method for an arithmetic encoding and decoding of binary states, wherein

in a first step a presetable value range for the specification of the interval width R is separated in K representative interval widths $\{Q_1, \ldots, Q_K\}$, a presetable value range for the specification of the probabilities is separated in N representative probability states $\{P_1, \ldots, P_N\}$ and allocation regulations are given, which allocate one Q_K ($1 \leq k \leq K$) to every interval width R and one P_n ($1 \leq n \leq N$) to every probability, and wherein in a second step the encoding or decoding of the binary states take place by performing the calculation of the new interval width to be derived in the encoding or decoding process, respectively, using a representative interval width Q_K ($1 \leq k \leq K$) and a representative probability state P_n ($1 \leq n \leq N$) by arithmetic operations other than multiplication and division, wherein the representative interval width Q_K is determined by the basic basis interval of the width R and the representative probability state P_n is determined by the probability estimation underlying the symbol to be encoded or to be decoded according to the given allocation regulations.

23. (original) A computer-readable storage medium on which a computer program is stored which enables a computer after it has been loaded into the storage of the computer to perform a method for an arithmetic encoding and decoding of binary states, wherein

in a first step a presetable value range for the specification of the interval width R is separated in K representative interval widths $\{Q_1, \ldots, Q_K\}$, a presetable value range for the specification of the probabilities is separated in N representative probability states $\{P_1, \ldots, P_N\}$ and allocation regulations are given, which allocate one Q_K ($1 \leq k \leq K$) to every interval width R and one P_n ($1 \leq n \leq N$) to every probability, and wherein in a second step the encoding or decoding of the binary states take place by performing the calculation of the new interval width to be derived in the encoding or decoding process, respectively, using a representative interval width Q_K ($1 \leq k \leq K$) and a representative probability state P_n ($1 \leq n \leq N$) by arithmetic operations other than multiplication and division, wherein the representative interval width Q_K is determined by the basic basis interval of the width R and the representative probability state P_n is determined by the probability estimation underlying the symbol to be encoded or to be decoded according to the given allocation regulations.

24. (original) The computer program according to claim 22, which is downloaded from an electronic data network, like for example from the internet, onto a data processing means which is connected to the data network.

--25. (new) A method for arithmetically encoding a symbol to be encoded having a binary state based on a current interval width R and a probability representing a probability estimation for the symbol to be encoded, wherein the probability is represented by a probability index for addressing a probability state from a plurality

of representative probability states, which method comprises the following steps:

encoding the symbol to be encoded by performing the following substeps:

mapping the current interval width to a quantization index from a plurality of representative quantization indices; and

performing the interval separation by accessing an interval division table using the quantization index and the probability index to obtain a partial interval width value.

26. (new) The method of claim 25, wherein the encoding further takes place by the following step:

updating the current interval width using the interval width value to obtain a new, updated interval width.

27. (new) The method of claim 25, wherein the partial interval width value specifies a width of a partial interval for a symbol to be encoded with a less probable state from a current interval with a current interval width.

28. (new) The method of claim 25, wherein updating the current interval width is further performed depending on the binary state of the symbol to be encoded.

29. (new) The method of claim 25, further comprising the following step:

adaptation of the probability estimation, wherein the adaptation of the probability estimation comprises looking up, with the probability index, in an LPS transition rule table (Next_State_LPS) to obtain a new probability index, when the symbol to be encoded has a

less probable state, and looking up, with the probability index, in an MPS transition rule table (Next_State_MPS) to obtain a new probability index, when the symbol to be encoded has a more probable state.

30. (new) The method of claim 29, further comprising adjusting a value indicative of the more probable state from a state originally indicated to the binary state of the symbol to be encoded, when the probability index is like a predetermined probability index and the symbol to be encoded has a binary state different from the state originally indicated.

31. (new) The method of claim 25, wherein the substep of updating the current interval width comprises the following steps:

equating the new interval width with the difference of current interval width minus the partial interval width value; and

subsequently, if the symbol to be encoded has a less probable state, equating the new interval width with the partial interval width value.

32. (new) The method of claim 25, wherein a current interval is represented by the current interval width and a current offset point, and the encoding is further performed by the following substep:

accumulating the current offset point and a difference of current interval width and partial interval width value to obtain a new, updated offset point, when the symbol to be coded has a less probable state.

33. (new) A method for arithmetically decoding an encoded symbol having a binary state based on a current interval width R and a probability representing a probability

estimation for the encoded symbol, wherein the probability is represented by a probability index of a probability state from a plurality of representative probability states, wherein the method comprises the following step:

decoding the encoded symbol by performing the following substeps:

mapping the current interval width to a quantization index from a plurality of representative quantization indices; and

performing the interval division by accessing an interval division table using the quantization index and the probability index to obtain a partial interval width value.

34. (new) The method of claim 33, wherein the decoding further takes place by means of the following step:

updating the current interval width using the partial interval width value to obtain a new, updated interval width.

35. (new) The method of claim 33, wherein the partial interval width value specifies a width of a partial interval for an encoded symbol with a less probable state from a current interval with the current interval width.

36. (new) The method of claim 33, wherein updating the current interval width is further performed depending on a value within a new partial interval characterized by the current partial interval width and the value within a new partial interval.

37. (new) The method of claim 36, wherein the decoding is further performed by means of the following substep:

equating the binary state of the encoded symbol with one of a more improbable and a more probable state depending on whether the value within the new partial interval is larger or smaller than a difference of the current interval width and partial interval width value.

38. (new) The method of claim 36, wherein the encoding is further performed by means of updating the value within the new partial interval with a next bit to be read in.

39. (new) The method of claim 36, further comprising the following step:

updating the probability estimation, wherein updating the probability estimation comprises looking up, with the probability index, in an LPS transition rule table (Next_State_LPS) to obtain a new probability index, when the value within the new partial interval is larger than a difference of the current interval width and partial interval width value, and looking up, with the probability index, in an MPS transition rule table (Next_State_MPS) to obtain a new probability index, when the value within the new partial interval is smaller than a difference of the current interval width and partial interval width value.

40. (new) The method of claim 36, further comprising adjusting a value indicative of the more probable state of the encoded symbol from a state originally indicated to a different binary state, when the probability index is like a predetermined probability index and the value within the new partial interval is larger than a difference of the current interval width and partial interval width value.

41. (new) The method of claim 33, wherein the current interval width is represented with an accuracy of b bits, and the

partial interval width value obtained from the interval division table is represented with an accuracy of b-2 bits.

42. (new)The method of claim 33, wherein

the substep of mapping comprises applying a shift and bit masking operation to a computer-internal/binary representation of the current interval width.

43. (new) The method of claim 33, wherein

the substep of mapping comprises applying a shift operation to a computer-internal/binary representation of the current interval width to obtain a quantized value for the current interval width, and a downstream access to a table (Qtab) to obtain the quantization index.

44. (new) The method of claim 33, wherein,

in the interval division table, values for the current interval width corresponding to all possible quantization indices and to all probability indices are filed as product values between quantization index, and in a table Rtab.

45. (new)The method of claim 33, further comprising the following step:

updating the probability estimation, wherein updating the probability estimation is performed by means of transition rules, wherein the transition rules specify which new probability state from a plurality of probability states, based on the symbol to be encoded and/or the encoded symbol, will be used for a next symbol to be encoded and/or an encoding symbol.

46. (new) The method of claim 33, further comprising the following step:

updating the probability estimation, wherein updating the probability estimation comprises looking up, with the probability index, in a transition rule table (Next_State_LPS) to obtain a new probability index.

47. (new) The method of claim 33, wherein

the number of possible quantization indices and/or the number of the probability states are selected depending on the preset accuracy of the coding and/or depending on the available storage room.

48. (new) The method of claim 33, further comprising the following substep:

renormalizing the new updated offset point and the new, updated interval width.

49. (new) The method of claim 33, wherein

decoding includes the following steps:

a) Determination of the LPS

b) Quantization of R:

q_index = Qtab [R>>q]

c) Determination of R_{LPS} and R:

R_{LPS} = Rtab [q_index, p_state]

R = R - R_{LPS}

d) Determination of bit, depending on the position of the partial interval:

if (V $\geq$ R) **then**

  bit $\leftarrow$ LPS

  V $\leftarrow$ V - R

R $\leftarrow$ R_{LPS}

if (p_state = 0) then valMPS $\leftarrow$ 1 - valMPS

p_state ← Next_State_LPS[p_state]

else

bit ← MPS

p_state ← Next_State_MPS[p_state]

e) Renormalization of R, reading out one bit and updating V,

wherein

q_index describes the index of a quantization value read out of Qtab,

p_state describes the current state,

R_{LPS} describes the interval width corresponding to the LPS,

ValMPS describes the bit corresponding to the MPS and

V describes a value from within the current partial interval.

50. (new) The method of claim 33, wherein,

in encoding and/or decoding, mapping to the quantization index q_index is performed according to the calculation regulation:

q_index = (R >> q) & Qmask

wherein Qmask represents a bit mask suitably selected depending on the number of probability states, R represents the current interval width and q represents a number of bits.

51. (new) The method of claim 33, wherein,

in the presence of a uniform probability distribution,

- in the encoding, the following calculation regulation is performed:

  R ← R >> 1

  if (bit = 1) **then**

  L ← L + R,

  or

  the following calculation regulation is performed:

L ← **L** << **1**
if (bit = 1) **then**
    L ← **L** + **R**
and, in the last alternative, a renormalization with doubled decision threshold values is performed and no doubling of L and R is carried out.

52. (new) The method of claim 33, wherein,

in the decoding, the following calculation regulation is performed:

R ← **R** >> **1**
if (V ≥ R) **then**
    bit ← **1**
    V ← **V** − **R**
else
    bit ← **0,**

or

the following calculation regulation:

m) Reading out one bit and updating V

n) Determination of bit depending on the position of the partial interval:

if (V ≥ R) **then**
    bit ← **1**
    V ← **V** − **R**
else
    bit ← **0.**

53. (new) The method of claim 33, wherein

the initialization of the probability models is performed depending on a quantization parameter SliceQP and preset model parameters m and n, wherein SliceQP describes the quantization parameter preset at the beginning of a slice, and m and n describe the model parameters.

54. (new) The method of claim 33, wherein

the initialization of the probability models includes the following steps:

```
k) preState = min(max(1, ((m * SliceQP) >>4)+n), 2*N)
l) if (preState <=N) then
      p_state = N - preState
      valMPS = 0
   else
      p_state = preState - (N+1)
      valMPS = 1,
```

wherein valMPS describes the bit corresponding to the MPS, SliceQP describes the quantization parameter preset at the beginning of a slice, and m and n describe the model parameters.

55. (new) The method of claim 33, wherein

the probability estimation of the states is performed by means of a finite state machine (FSM).

56. (new) The method of claim 33, wherein

the generation of the probability states is performed offline.

57. (new) The method of claim 33, wherein

the selection of the states depends on the statistics of the data to be coded and/or on the number of the states.

58. (new) An arrangement for arithmetically encoding a symbol to be encoded having a binary state based on a current interval width R and a probability representing a probability estimation for the symbol to be encoded, wherein the probability is represented by a probability index for addressing a probability state from a plurality of representative probability states, the device comprising:

means for encoding the symbol to be encoded, including the following means:

means for mapping the current interval width to a quantization index from a plurality of representative quantization indices; and

means for performing the interval separation by accessing an interval division table using the quantization index and the probability index to obtain a partial interval width value.

59. (new) An arrangement for arithmetically decoding an encoded symbol having a binary state based on a current interval width R and a probability representing a probability estimation for the encoded symbol, wherein the probability is represented by a probability index for addressing a probability state from a plurality of representative probability states, the device comprising:

means for decoding the encoded symbol, comprising the following means:

means for mapping the current interval width to a quantization index from a plurality of representative quantization indices; and

means for performing the interval separation by accessing an interval division table using the quantization index and the probability index to obtain a partial interval width value.

60. (new) A computer program which enables a computer after it has been loaded into the storage of the computer to perform a method for arithmetically encoding a symbol to be encoded having a binary state based on a current interval width R and a probability representing a probability estimation for the symbol to be encoded,

wherein the probability is represented by a probability index for addressing a probability state from a plurality of representative probability states, the method comprising the following steps:

encoding the symbol to be encoded by performing the following sub-steps:

mapping the current interval width to a quantization index from a plurality of representative quantization indices; and

performing the interval separation by accessing an interval division table using the quantization index and the probability index to obtain a partial interval width value.

61. (new) A computer program which enables a computer after it has been loaded into the storage of the computer to perform a method for arithmetically decoding an encoded symbol having a binary state based on a current interval width R and a probability representing a probability estimation for the encoded symbol, wherein the probability is represented by a probability index of a probability state from a plurality of representative probability states, the method comprising the following steps:

decoding the encoded symbol by performing the following sub-steps:

mapping the current interval width to a quantization index from a plurality of representative quantization indices; and

performing the interval division by accessing an interval division table using the quantization index

and the probability index to obtain a partial interval width value.

62. (new) A computer-readable storage medium on which a program is stored which enables a computer after it has been loaded into the storage of the computer to perform a method for arithmetically encoding a symbol to be encoded having a binary state based on a current interval width R and a probability representing a probability estimation for the symbol to be encoded, wherein the probability is represented by a probability index for addressing a probability state from a plurality of representative probability states, the method comprising the following steps:

encoding the symbol to be encoded by performing the following sub-steps:

mapping the current interval width to a quantization index from a plurality of representative quantization indices; and

performing the interval separation by accessing an interval division table using the quantization index and the probability index to obtain a partial interval width value.

63. (new) A computer-readable storage medium on which a program is stored which enables a computer after it has been loaded into the storage of the computer to perform a method for arithmetically decoding an encoded symbol having a binary state based on a current interval width R and a probability representing a probability estimation for the encoded symbol, wherein the probability is represented by a probability index of a probability state from a plurality of representative probability states, the method comprising the following steps:

decoding the encoded symbol by performing the following sub-steps:

mapping the current interval width to a quantization index from a plurality of representative quantization indices; and

performing the interval division by accessing an interval division table using the quantization index and the probability index to obtain a partial interval width value.

64. (new) The computer-readable storage medium on which a program is stored which enables a computer after it has been loaded into the storage of the computer to perform a method according to claim 60.

65. (new) The computer-readable storage medium on which a program is stored which enables a computer after it has been loaded into the storage of the computer to perform a method according to claim 61.

--